Fast Learning for Non-Parallel Many-to-Many Voice Conversion with Residual Star Generative Adversarial Networks

Shengkui Zhao, Trung Hieu Nguyen, Hao Wang, Bin Ma

Machine Intelligence Technology, Alibaba Group

{shengkui.zhao, trunghieu.nguyen, hao.w, b.ma}@alibaba-inc.com

Abstract

This paper proposes a fast learning framework for non-parallel many-to-many voice conversion with residual Star Generative Adversarial Networks (StarGAN). In addition to the state-of-the-art StarGAN-VC approach that learns an unreferenced mapping between a group of speakers' acoustic features for non-parallel many-to-many voice conversion, our method, which we call Res-StarGAN-VC, presents an enhancement by incorporating a residual mapping. The idea is to leverage on the shared linguistic content between source and target features during conversion. The residual mapping is realized by using identity shortcut connections from the input to the output of the generator in Res-StarGAN-VC. Such shortcut connections accelerate the learning process of the network with no increase of parameters and computational complexity. They also help generate high-quality fake samples at the very beginning of the adversarial training. Experiments and subjective evaluations show that the proposed method offers (1) significantly faster convergence in adversarial training and (2) clearer pronunciations and better speaker similarity of converted speech, compared to the StarGAN-VC baseline on both mono-lingual and cross-lingual many-to-many voice conversion tasks.

Index Terms: Voice conversion (VC), non-parallel VC, many-to-many VC, generative adversarial networks (GANs), StarGAN-VC, Res-StarGAN-VC

1. Introduction

The primary goal of voice conversion (VC) is to convert the voice of a source speaker to that of a target speaker, while having the same linguistic content as the original sample. There are many application scenarios of VC systems, such as speech enhancement [1, 2], speaking-assistance [3, 4], and personalized text-to-speech (TTS) systems [5].

VC may be categorized into mono-lingual or cross-lingual tasks according to whether source and target speakers speak the same language. Parallel data (i.e. both source and target speakers utter the same sentences) may be collected for mono-lingual VC. Many existing effective approaches [6, 7, 8, 9, 10, 11] based on parallel data demonstrated good performance for mono-lingual VC tasks, such as the Gaussian mixture models (GMMs) based methods [6, 7], the neural network (NN) based methods [8, 9] and the non-negative matrix factorization (NMF) based methods [10, 11]. However, parallel data collection is not possible for cross-lingual VC and the requirement of parallel training data greatly limits the usability of the above approaches in practical scenarios.

In the past few years, there has been growing interest in non-parallel VC. The Voice Conversion Challenge 2018 (VCC 2018) has further boosted the development of non-parallel approaches [12]. Especially, the non-parallel system "N10" [13] achieved good results for the non-parallel many-to-many *Spoke* subtask. However, this system has the following limitations: (1) it depends heavily on large amounts of automatic speech recognition and text-to-speech corpora; (2) it is a many-to-one VC system per se. Each target speaker needs to have its own model trained. The VC frameworks based on conditional variational autoencoders (CVAEs) [14, 15] can work on limited training data, but suffer from over-smoothness problem in the outputs. Generative adversarial networks (GANs) [16] have been considered as powerful framework to alleviate the weakness of the CVAE-based VC framework [17, 18]. GANs consists of two adversarial networks that compete with each other in training: a generator tries to generate new samples to fool a discriminator and the discriminator tries to distinguish the generated samples from the real samples. As an import variant of GANs, StarGAN [19] supports many-to-many domain mapping using a single model. In StarGAN, the generator takes in as inputs both feature and domain information, and learns to flexibly translate the input feature into the corresponding domain. Label of the input feature is used to represent the domain information. Inheriting all advantages of StarGAN, the recent proposed StarGAN-VC [18] supports many-to-many VC tasks trained across different attribute domains of speakers with no requirement of parallel data, and it has demonstrated promising results even with limited training utterances.

While StarGAN-VC provides a promising framework for non-parallel many-to-many VC, the learning process of StarGAN-VC is slow, and the converted audio is of insufficient speech quality. One reason is that StarGAN-VC is to learn a direct unreferenced mapping between a group of speakers' acoustic features. On non-parallel data, learning such a mapping becomes inefficient since the network learns both conversion of speaker identity and preservation of linguistic content together. Inspired by the deep residual learning framework [20] that uses shortcut connections between network layers to learn referenced residual mappings for easing the optimization of deeper neural networks, we propose to learn a residual mapping leveraged on the shared linguistic content between source and target speakers' speech during conversion by adding identity shortcut connections straight from the input to the output of the generator. We will demonstrate that the introduction of such shortcut connections not only accelerates the learning process of the network with no increase of parameters and computational complexity but also helps generate high-quality fake samples at the very beginning of the adversarial training. We call our proposed method Res-StarGAN-VC. Experiments and subjective evaluations show that Res-StarGAN-VC offers (1) significantly faster convergence in adversarial training and (2) clearer pronunciations and better speaker similarity of converted speech, compared to the StarGAN-VC baseline, on both mono-lingual and cross-lingual many-to-many VC tasks.

The remainder of this paper is organized as follows. We firstly introduce the StarGAN-VC and then propose our residual

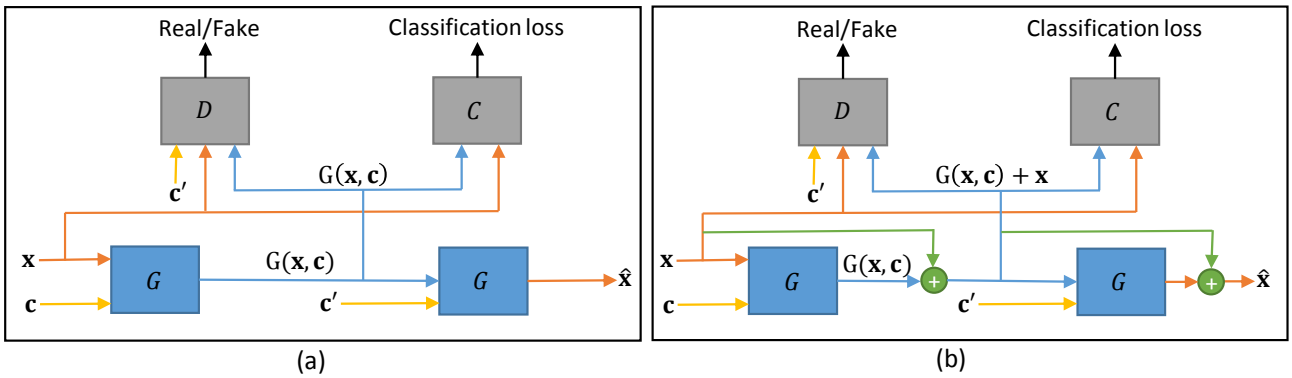

Figure 1: *Schematic diagrams of (a) StarGAN-VC, and (b) Res-StarGAN-VC.*

learning framework for VC and Res-StarGAN-VC in Section 2. The experimental results are given in Section 3. Section 4 concludes this paper.

2. Residual StarGAN Voice Conversion

2.1. The Previous StarGAN-VC Approach

Let us consider $H(\mathbf{x})$ as an underlying VC mapping from $\mathbf{x}$ to $\mathbf{y}$ with $\mathbf{x}$ and $\mathbf{y}$ denoting source and target acoustic feature sequences, respectively. It is assumed $\mathbf{x}$ and $\mathbf{y}$ have the same dimensions. If this is not the case, a linear projection is applied to $\mathbf{x}$ to match the dimensions of $\mathbf{y}$. StarGAN-VC learns a generator $G(\mathbf{x})$ to asymptotically approximate $H(\mathbf{x})$ as shown in Fig. 1(a). It consists of a generator G, a discriminator D, and a classifier C. D aims to distinguish between converted and real speech features by probabilities and C is used to classify the converted feature $G(\mathbf{x})$ to the class corresponding to the target attribute $\mathbf{c}$. The attribute label $\mathbf{c}'$ and $\mathbf{c}$ represents the source and target speaker's attribute in training data, respectively. Depending on the conversion requirements, the attribute label $\mathbf{c}'$ and $\mathbf{c}$ can comprise one or more categories such as speaker identity, gender and language. Each category consists of two or more classes to divide speakers into different domains. In StarGAN, the classes are represented by using one-hot vectors and the attribute labels are formulated by concatenating all the one-hot vectors.

2.2. The Proposed Residual Learning Framework for VC

For the underlying VC mapping problem described above, if we can hypothesize that a generator $G(\mathbf{x})$ can asymptotically approximate $H(\mathbf{x})$ then it is equivalent to hypothesizing that the generator $G(\mathbf{x})$ can asymptotically approximate the residual mapping $H(\mathbf{x}) - \mathbf{x}$. Considering the shared linguistic information between $\mathbf{x}$ and $\mathbf{y}$, rather than directly use the generator $G(\mathbf{x})$ to approximate $H(\mathbf{x})$, we explicitly let the generator approximate $H(\mathbf{x}) - \mathbf{x}$. Then the mapping $H(\mathbf{x})$ thus becomes $H(\mathbf{x}) := G(\mathbf{x}) + \mathbf{x}$. The formulation of $G(\mathbf{x}) + \mathbf{x}$ can be realized by a shortcut connection (i.e. an identity mapping) from the input to the output of $G(\mathbf{x})$. We expect that mapping $G(\mathbf{x}) := H(\mathbf{x}) - \mathbf{x}$ is at least as simple as mapping $H(\mathbf{x})$ if not simpler. The identity shortcut connections add neither extra parameter nor computational complexity and the generator can still be trained as usual using stochastic gradient descent (SGD). The residual learning framework is not new and it has been used by K. He [20] to ease the optimization of deeper neural networks, which leads to the popular residual networks (ResNets). Our residual learning framework to train the generator has no constrain on its network architecture. The generator can still be configured with ResNets and we found the two residual learning frameworks work complementarily. Another related work is the input-to-output highway networks proposed in [9] where the input of a DNN-based model is connected to its output for converting speech parameters. Different from the DNN-based highway networks, we apply the residual learning framework and adversarial training for GAN-based model. During the adversarial training for the generator, the added shortcut connection can bypass its input to its output directly which is likely to provide high quality fake samples from the start of training and drives a fast learning of the discriminator for avoiding more mistakes. This in turn leads to the fact that the generator also need to learn fast to generate better fake samples to fool the discriminator. In consequence, the whole adversarial training is driven efficiently and reaches the equilibrium point quickly.

2.3. The Proposed Res-StarGAN-VC Approach

With the above described residual learning framework, let us consider to train a generator G such that $G(\mathbf{x}) + \mathbf{x}$ maps source acoustic feature sequence $\mathbf{x}$ into target acoustic feature sequence $\mathbf{y}$, where $\mathbf{y}$ is conditioned on a target attribute label $\mathbf{c}$ as illustrated in Fig. 1(b). In another words, our proposed Res-StarGAN-VC trains the generator G to approximate the residual sequences: $G(\mathbf{x}, \mathbf{c}) \rightarrow \mathbf{y} - \mathbf{x}$. The goal of training G is to make the converted feature $G(\mathbf{x}) + \mathbf{x}$ as realistic as real speech features while preserving the target speaker's voice characteristics. To achieve this, Res-StarGAN-VC applies a discriminator D to distinguish between converted and real speech features by probabilities and also applies a classifier C to classify the converted feature $G(\mathbf{x}) + \mathbf{x}$ to the class corresponding to the target attribute $\mathbf{c}$. Compared to the configuration of StarGAN-VC in Fig. 1(a), two shortcut connections are added between the input and the output of the generator G in Res-StarGAN-VC, by which we add the inputs of G to the outputs of G. The training losses for Res-StarGAN-VC are described as follows.

Adversarial Loss: To train the generator G and the discriminator D, we define the adversarial loss as

$$\begin{aligned}\mathcal{L}_{adv} = & E_{\mathbf{x},\mathbf{c}'}[\log D(\mathbf{x}, \mathbf{c}')] \\ & + E_{\mathbf{x},\mathbf{c}}[\log(1 - D(G(\mathbf{x}, \mathbf{c}) + \mathbf{x}, \mathbf{c}))] \end{aligned} \quad (1)$$

where $E[\cdot]$ represents expectation operation; $G(\mathbf{x}, \mathbf{c})$ represents the residual mapping conditioned on the input $\mathbf{x}$ and target

speaker's attribute label $\mathbf{c}$; $\mathbf{c}'$ is source speaker's attribute label. $G(\mathbf{x}, \mathbf{c}) + \mathbf{x}$ is the converted output for a target speaker with label $\mathbf{c}$. During training, the real target output is not available. We feed $\mathbf{x}$ with its label $\mathbf{c}'$ to D as real acoustic feature sequence and feed $G(\mathbf{x}, \mathbf{c}) + \mathbf{x}$ with label $\mathbf{c}$ as fake sample. D aims to distinguish between $(\mathbf{x}, \mathbf{c}')$ and $(G(\mathbf{x}, \mathbf{c}) + \mathbf{x}, \mathbf{c})$. In the adversarial training, G tries to minimize this adversarial loss, while D tries to maximize it.

Domain Classification Loss: The classifier C is used to optimize G such that the converted output $G(\mathbf{x}, \mathbf{c}) + \mathbf{x}$ can be classified to the class corresponding to the target attribute label $\mathbf{c}$. C is trained by the real sample $(\mathbf{x}, \mathbf{c}')$ to minimize the following classification loss

$$\mathcal{L}_{cls}^{C} = E_{\mathbf{x},\mathbf{c}'}[-\log C(\mathbf{c}'|\mathbf{x})] \tag{2}$$

where the term $C(\mathbf{c}'|\mathbf{x})$ represents a probability distribution over the attribute label $\mathbf{c}'$. By minimizing this objective, C learns to classify a real sample x to its corresponding label $\mathbf{c}'$. When we feed the converted output $G(\mathbf{x}, \mathbf{c}) + \mathbf{x}$ to C, it produces the following classification loss

$$\mathcal{L}_{cls}^{G} = E_{\mathbf{x},\mathbf{c}}[-\log C(\mathbf{c}|(G(\mathbf{x}, \mathbf{c}) + \mathbf{x}))] \tag{3}$$

where G tries to minimize this loss to achieve high classification accuracy for the target attribute $\mathbf{c}$. StarGAN-VC makes a change from StarGAN that instead of merging C and D to the same network it separates C and D into different networks. This allows C to be pretrained. Res-StarGAN-VC follows StarGAN-VC to separate C and D.

Reconstruction Loss: To guarantee that the linguistic content are preserved while converting the speaker acoustic identity during conversion, a cycle consistency loss is also applied to G, which is defined as

$$\begin{aligned}\mathcal{L}_{rec} &= E_{\mathbf{x},\mathbf{c},\mathbf{c}'}[||\mathbf{x} - (G(G(\mathbf{x}, \mathbf{c}) + \mathbf{x}, \mathbf{c}') \\ &\quad + (G(\mathbf{x}, \mathbf{c}) + \mathbf{x}))||_1] \\ &= E_{\mathbf{x},\mathbf{c},\mathbf{c}'}[||G(G(\mathbf{x}, \mathbf{c}) + \mathbf{x}, \mathbf{c}') + G(\mathbf{x}, \mathbf{c})||_1]\end{aligned} \tag{4}$$

where G takes the converted feature $G(\mathbf{x}, \mathbf{c}) + \mathbf{x}$ and the source attribute label $\mathbf{c}'$ as input and tries to reconstruct the source $\mathbf{x}$. L_1 norm is adopted for the reconstruction loss. To ensure an identity mapping for the same source and target speaker, the following identity mapping loss is also applied

$$\begin{aligned}\mathcal{L}_{id} &= E_{\mathbf{x},\mathbf{c}'}[||(G(\mathbf{x}, \mathbf{c}') + \mathbf{x}) - \mathbf{x}||_1] \\ &= E_{\mathbf{x},\mathbf{c}'}[||G(\mathbf{x}, \mathbf{c}')||_1]\end{aligned} \tag{5}$$

where $G(\mathbf{x}, \mathbf{c}')$ is considered as a zero mapping.

Full Objective: The full objective functions of Res-StarGAN-VC can be summarized as follows

$$\mathcal{L}_D = -\mathcal{L}_{adv}, \tag{6}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{G} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{id}\mathcal{L}_{id}, \tag{7}$$

$$\mathcal{L}_C = \mathcal{L}_{cls}^{C} \tag{8}$$

where λ_{cls}, λ_{rec}, and λ_{id} are hyper-parameters to control the weights of the corresponding losses relative to the adversarial loss. We use $\lambda_{cls} = 10$, $\lambda_{rec} = 10$, and $\lambda_{id} = 10$ in our experiments.

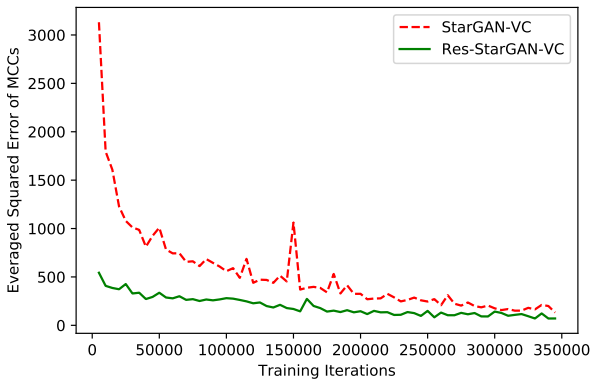

Figure 2: *Comparison of average squared errors of the converted MCCs over training iterations.*

3. Experiments

3.1. Experimental Setup

Our Res-StarGAN-VC is implemented similarly as StarGAN-VC described in [18]. In Res-StarGAN-VC, all networks of G, D, and C use the same network structures as StarGAN-VC. The acoustic feature sequence is treated as an image of size $l \times d$ where l denotes number of speech frames and d denotes speech feature dimension per frame. The gated 2D CNNs [21] are used to construct G, D and C. The entire architecture is fully convolutional with no full-connected layers. This allows arbitrary lengths of input sequences during conversion. All parameter settings are kept the same to ensure that Res-StarGAN-VC introduces neither extra parameter nor computational complexity compared to StarGAN-VC.

We conduct two experiments to evaluate our method on non-parallel many-to-many mono-lingual and cross-lingual VC tasks. The first experiment is conducted on mono-lingual VC task and the VCC 2018 dataset [12] is used. In the dataset, there are four source speakers and four target speakers of balanced genders. All speakers are native US English speakers. The training set consists of 81 sentences for each speaker. The evaluation set consists of 35 sentences for each source speaker. All 16 source-target conversion pairs are evaluated. The second experiment is a cross-lingual VC task. In this experiment, we use speech data from 10 English speakers and 10 Mandarin speakers. 7 speakers are from CMU-ARCTIC dataset [22]. The other 3 English speakers and 10 Mandarin speakers are from internal dataset of Alibaba. For all speakers, we randomly select 1100 sentences for training and 130 sentences for test. Both cases of English-to-Mandarin conversion and Mandarin-to-English conversion are evaluated. For each case, 20 sentences are randomly selected for listening test. All the speech signals are resampled at 16kHz/16bit for system training and evaluation.

In our experiments, the acoustic feature sequences are 36 ($d = 36$) mel-cepstral coefficients (MCCs). The MCCs are extracted from spectral envelopes windowed with 25 ms and shifted every 5 ms using WORLD analyzer [23]. The length of each acoustic feature sequence is 512 ($l = 512$) frames. The logarithmic fundamental frequency (log F0) contours were converted using the logarithm Gaussian normalized transformation [24]. The converted MCCs, F0, together with the original aperiodicities (APs) are synthesized to time-domain waveform using WORLD vocoder. All algorithms were implemented using

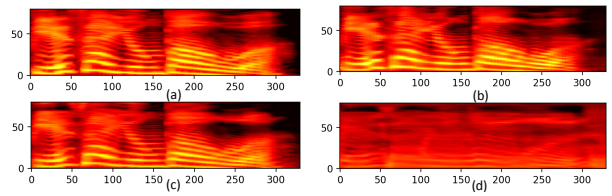

Figure 3: *Example plots of mel-spectrograms of (a) source natural input, (b) output of StarGAN-VC, (c) output of Res-StarGAN-VC, (d) $G(\mathbf{x})$ of Res-StarGAN-VC.*

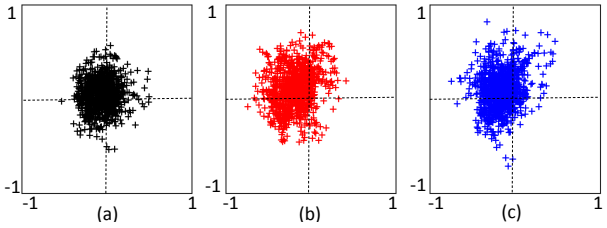

Figure 4: *Example scatter plots of the 15th MCC via the 20th MCC of (a) source natural input, (b) output of StarGAN-VC, (c) output of Res-StarGAN-VC*

PyTorch [25]. During training, the target attribute label $\mathbf{c}$ is randomly generated.

3.2. Performance Evaluation

3.2.1. Learning Speed

We first compared the learning speed of StarGAN-VC and Res-StarGAN-VC. To evaluate the learning speed, both methods are trained for 350,000 iterations where they both reach convergence states. For each 5,000 iteration interval, we perform the conversions for all the evaluation sets and store the converted MCCs. We use the converted results at 350,000 iterations as the final results for both methods and compute the average squared errors of MCCs between every previous stored results and the final results. The average squared errors over all evaluation sets of the two methods are shown in Fig. 2. The reduction speed of the average squared errors indicates the learning speeds of the methods. From Fig. 2, we can see that Res-StarGAN-VC provides a faster learning speed than StarGAN-VC. Res-StarGAN-VC always provide smaller average squared errors than StarGAN-VC during training process.

3.2.2. Learned Results

Fig. 3 shows plots of mel-spectrograms extracted from waveforms of the audio speech after convergence of training. Fig. 3(a) illustrates plot for speech recording. Fig. 3(b) is the plot for the synthesized speech by StarGAN-VC, which is the unreferenced mapping output of G. Fig. 3(c) is the plot for the synthesized speech by Res-StarGAN-VC, which is the sum of the input of G and the learned residual by G. Fig. 3(d) is the learned residual signal from G of Res-StarGAN-VC. We can see that both methods are able to learn well the feature structure. But StarGAN-VC tends to produce more noise distortions on the converted features. The use of input feature in Res-StarGAN-VC effectively alleviates the noise distortions. Compared to Fig. 3(b) where both the speaker's identity and linguistic information have been learned, Fig. 3(d) shows that the generator G of Res-StarGAN-VC only learns the residual difference between the target speaker's feature sequence and the source speaker's feature sequence.

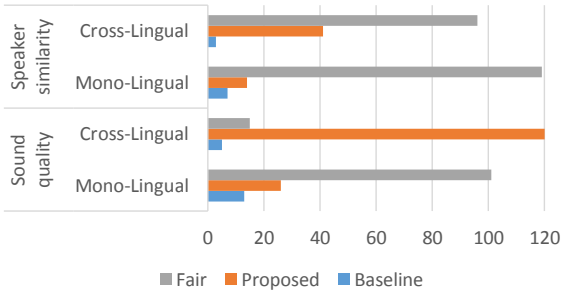

Figure 5: *Results of the preference test of sound quality and speaker similarity*

In Fig. 4, we illustratively show that both StarGAN-VC and Res-StarGAN-VC can effectively alleviate the over-smoothing problem occurred in many other VC approaches [9, 14, 15]. Fig. 4(a), (b), and (c) plot pairs of MCCs of the input recording, the output of StarGAN-VC, and the output of Res-StarGAN-VC. From Fig. 4, we can see that both VC approaches are effective to reproduce the characteristics of natural speech.

3.2.3. Subjective Test

The subjective evaluation is used to compare Res-StarGAN-VC with StarGAN-VC on both mono-lingual and cross-lingual VC tasks. Preference AB and ABX tests are conducted to compare their performance in terms of speech quality and speaker similarity. In both tests, 20 sentences are randomly selected for both experiments. 7 experienced listeners are invited to participate in the listening tests. In AB test, the listeners are asked to select 'A', 'B' or 'Fair' based the sound quality of the presented A and B audio. In the ABX test, the target speaker's speech are used as the reference 'X'. The listeners are asked to select 'A', 'B' or 'Fair' based the similarity of the presented A and B audio files referenced to 'X'. The results of the preference tests on sound quality and speaker similarity for both mono-lingual and cross-lingual VCs are shown in Fig. 5. We find that our proposed Res-StarGAN-VC scores higher in both speech quality and speaker similarity. A greater improvement is observed for cross-lingual task compared with the mono-lingual task.

4. Conclusions

This paper presented a fast learning framework using residual Star Generative Adversarial Networks (Res-StarGAN-VC) for the non-parallel many-to-many voice conversion task. Unlike StarGAN-VC which learns an unreferenced mapping, Res-StarGAN-VC learns a residual mapping and dramatically improves the learning speed in the training process. The added identity mapping in Res-StarGAN-VC allows the direct use of the input of the generator for estimating the converted speech feature. Subjective evaluation shows that the sound quality and speaker similarity are improved. Our experimental studies also reveal that Res-StarGAN-VC has potential to be simplified. We will investigate these problems in our future work.

5. References

[1] Z. Inanoglu and S. Young, "Data-driven emotion conversion in spoken english," *Speech Communications*, vol. 51, no. 3, pp. 268–283, 2009.

[2] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 9, pp. 2505–2517, 2012.

[3] A. B. Kain, J.-P. Hosom, X. Niu, J. P. van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," *Speech Communications*, vol. 49, no. 9, pp. 743–759, 2007.

[4] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communications*, vol. 54, no. 1, pp. 134–146, 2012.

[5] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998, p. 285–288.

[6] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximumlikelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, p. 2222–2235, 2007.

[7] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 5, p. 912–921, 2010.

[8] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 5, p. 954–964, 2010.

[9] Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using input-to-output highway networks," *IEICE Transactions on Information and Systems*, vol. E100-D, no. 8, p. 1925–1928, 2017.

[10] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Transactions on Information and Systems*, vol. E96-A, no. 10, p. 1946–1953, 2013.

[11] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 10, p. 1506–1521, 2014.

[12] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv:1804.04262*, Apr. 2018.

[13] L. Liu, Z. Ling, Y. Jiang, M. Zhou, and L. Dai, "Wavenet vocoder with limited training data for voice conversion," in *Proc. INTERSPEECH*, 2018, pp. 1983–1987.

[14] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Processing Association Annual Summit and Conference (APSIPA ASC)*, 2016, pp. 1–6.

[15] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, p. 5274–5278.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. Neural Information Processing Systems (NIPS)*, 2014, p. 2672–2680.

[17] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv:1711.11293*, Nov. 2017.

[18] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv:1806.02169v2*, Jun. 2018.

[19] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multidomain image-to-image translation," *arXiv:1711.09020 [cs.CV]*, Nov 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. International Conference on Machine Learning (ICML)*, 2017, p. 933–941.

[22] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *5th ISCA Speech Synthesis Workshop*, 2004.

[23] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99-D, no. 7, p. 1877–1884, 2016.

[24] K. Liu, J. Zhang, and Y. Yan, "High quality voice conversion through phoneme-based linear mapping functions with straight for mandarin," in *Proc. International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2007, p. 410–414.

[25] "pytorch," https://github.com/pytorch/pytorch.